SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)
Coimbatore – 641 035.

Department of IT & AIML

19CST202 – Database Management Systems
Internal Examination I Important Questions

PART A

Define the purpose of DBMS

- It is a collection of programs that enables the user to create and maintain a database.
- Database Management Systems (DBMS) are software systems used to store, retrieve, and run queries on data.
- A DBMS serves as an interface between an end-user and a database, allowing users to create, read, update, and delete data in the database.

List out the types of DBMS.

The types of DBMS based on data model are as follows −

- Relational database.
- Object oriented database.
- Hierarchical database.
- Network database.

Define Network database.

A network database is a type of database model wherein multiple member records or files can be linked to multiple owner files and vice versa. The model can be viewed as an upside-down tree where each member information is the branch linked to the owner, which is the bottom of the tree.

Recall Project operation with an example or State about PROJECT operation in Relational algebra?

Project operation selects (or chooses) certain attributes discarding other attributes. The Project operation is also known as vertical partitioning since it partitions the relation or table vertically discarding other columns or attributes.

Notation: πA(R)

where

- 'A' is the attribute list, it is the desired set of attributes from the attributes of relation(R),
- symbol 'π(pi)' is used to denote the Project operator,
- R is generally a relational algebra expression, which results in a relation.

Example:

Π Class, Dept(Faculty)

Class	**Dept**
5	CSE
6	EE

Define Relational data model.

- Relational data model is the primary data model, which is used widely around the world for data storage and processing.
- This model is simple and it has all the properties and capabilities required to process data with storage efficiency.
- The relational model represents how data is stored in Relational Databases. A relational database stores data in the form of relations (tables).

Define instances and schemas of database?

Schema refers to the overall description of any given database.

Instance basically refers to a collection of data and information that the database stores at any particular moment.

Compare database systems with file systems. What are the disadvantages of file processing system

Basis	File System	DBMS
Structure	The file system is software that manages and organizes the files in a storage medium within a computer.	DBMS is software for managing the database.
Data Redundancy	Redundant data can be present in a file system.	In DBMS there is no redundant data.
Backup and Recovery	It doesn't provide backup and recovery of data if it is lost.	It provides backup and recovery of data even if it is lost.
Query processing	There is no efficient query processing in the file system.	Efficient query processing is there in DBMS.
Consistency	There is less data consistency in the file system.	There is more data consistency because of the process of normalization.
Complexity	It is less complex as compared to DBMS.	It has more complexity in handling as compared to the file system.
Security Constraints	File systems provide less security in comparison to DBMS.	DBMS has more security mechanisms as compared to file systems.
Cost	It is less expensive than DBMS.	It has a comparatively higher cost than a file system.
Data Independence	There is no data independence.	In DBMS data independence exists.

Basis	File System	DBMS
User Access	Only one user can access data at a time.	Multiple users can access data at a time.
Meaning	The user has to write procedures for managing databases	The user not required to write procedures.
Sharing	Data is distributed in many files. So, not easy to share data	Due to centralized nature sharing is easy
Data Abstraction	It give details of storage and representation of data	It hides the internal details of Database
Integrity Constraints	Integrity Constraints are difficult to implement	Integrity constraints are easy to implement
Example	Cobol, C++	Oracle, SQL Server

Disadvantages of File Processing System:

- Slow access time – Direct access of files is very difficult and one needs to know the entire hierarchy of folders to get to a specific file. This involves a lot of time.
- Presence of redundant data – The same data can be present in two or more files which takes up more disc space
- Inconsistent Data – Due to data redundancy, same data stored at different places might not match to each other
- Data Integrity Problems – The data present in the database should be consistent and correct. To achieve this, the data should must satisfy certain constraints. Difficulty in recovery of corrupt data – Recovery or backup of lost and corrupt data is nearly impossible in case of File Processing System.
- Lack of Atomicity –Operations performed in the database must be atomic i.e. either the operation takes place as a whole or does not take place at all.
- Problem in Concurrent Access – When a number of users operates on a common data in database at the same time then anomalies arise, due to lack of concurrency control.
- Unauthorized Access – Anyone who gets access to the file can read or modify the data.

Define high level/conceptual data model.

High-level conceptual data models provide concepts for presenting data in ways that are close to the way people perceive data. A typical example is the entity relationship model, which uses main concepts like entities, attributes and relationships.

What are the advantages and disadvantages of relational model.

Advantages

- Simple and Easy To Use - Storing data in tables is much easier to understand and implement as compared to other storage techniques.
- Manageability - Because of the independent nature of each relation in a relational database, it is easy to manipulate and manage. This improves the performance of the database.
- Query capability - With the introduction of relational algebra, relational databases provide easy access to data via high-level query language like SQL.
- Data integrity - With the introduction and implementation of relational constraints, the relational model can maintain data integrity in the database.

Disadvantages

- The performance of the relational model depends upon the number of relations present in the database.
- Hence, as the number of tables increases, the requirement of physical memory increases.
- The structure becomes complex and there is a decrease in the response time for the queries.
- Because of all these factors, the cost of implementing a relational database increase.

PART B – 13 Marks

1. i)Elaborate the three Schema architecture of a database management system with neat diagram.

The three-schema architecture divides the database into three-level used to create a separation between the physical database and the user application. In simple terms, this architecture hides the details of physical storage from the user.

The database administrator (DBA) responsible is to change the structure of database storage without affecting the user's view. It deals with the data, the relationship between them and the different access methods implemented on the database. The logical design of database is called a schema

This architecture contains three layers of database management system, which are as follows −

- External level
- Conceptual level
- Internal level

External/ View level

This is the highest level of database abstraction. It includes a number of external schemas or user views. This level provides different views of the same database for a specific user or a group of users. An external view provides a powerful and flexible security mechanism by hiding the parts of the database from a particular user.

Conceptual or Logical level

This level describes the structure of the whole database. It acts as a middle layer between the physical storage and user view. It explains what data to be stored in the database, what the data types are, and what relationship exists among those data. There is only one conceptual schema per database.

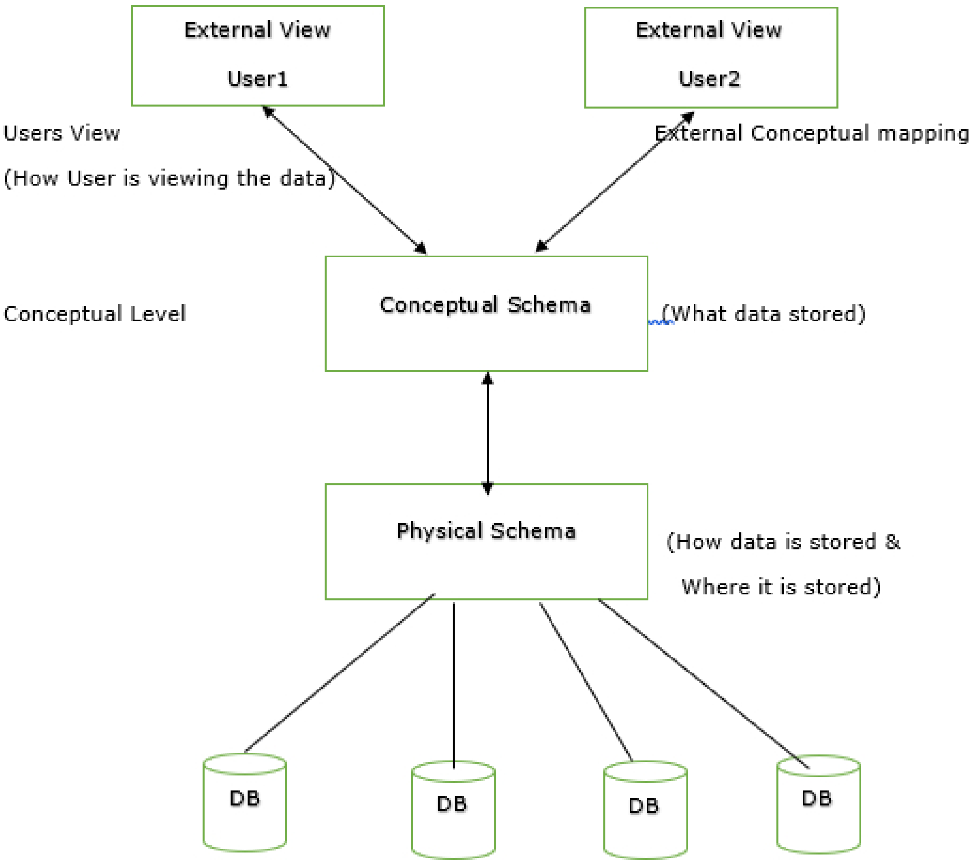
This level describes the structure of the whole database. It acts as a middle layer between the physical storage and user view. It explains what data to be stored in the database, what the data types are, and what relationship exists among those data. There is only one conceptual schema per database.

Internal or Physical level

This is the lowest level of database abstraction. It describes how the data is stored in the database and provides the methods to access data from the database. It allows viewing the physical representation of the database on the computer system.

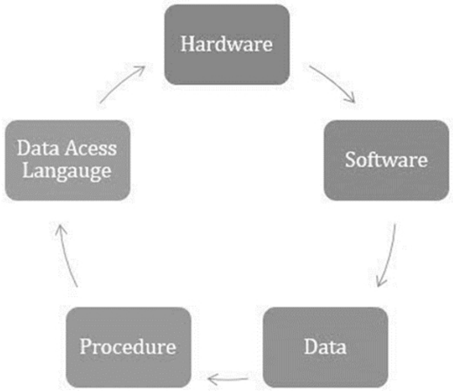
The interface between the conceptual and internal schema identifies how an element in the conceptual schema is stored and how it may be accessed. It is one which is closest to physical storage. The internal schema not only defines different stored record types, but also specifies what indices exist, how stored fields are represented.

The three-level schema architecture in DBMS is given below −



ii)Explain the components of Database Management System

Database environment is a collection of components that regulates the use of data, management, and a group of data. These components consist of people, the technique of Handel the database, data, hardware, software, etc. there are several components available for the DBMS.



'

Hardware

The hardware is the actual computer system used for keeping and accessing the database. The conventional DBMS hardware consists of secondary storage devices such as hard disks. Databases run on the range of machines from microcomputers to mainframes.

Software

Software is the actual DBMS between the physical database and the users of the system. All the requests from the user for accessing the database are handled by DBMS.

Data

It is an important component of the database management system. The main task of DBMS is to process the data. Databases are used to store the data, retrieved, and updated to and from the databases.

Users

There are a number of users who can access or retrieve the data on demand using the application and the interfaces provided by the DBMS.

The users of the database can be classified into different groups −

- Native Users
- Online Users
- Sophisticated Users
- Specialized Users
- Application Users
- DBA- Database Administrator

Procedure

- The main purpose of the procedure is to guide the user during the management and operation of the database.
- The procedure of the databases is so similar to the function of the database. The major difference between the database procedure and database function is that the database function acts the same as the SQL statement. In contrast, the database procedure is invoked using the CALL statement of the DBMS.
- Database procedures can be created in two ways in enterprise architecture. These two ways are as below.
    - The individual object or the default object.
    - The operations in a container.

Database Access Language

- Database Access Language is a simple language that allows users to write commands to perform the desired operations on the data that is stored in the database.
- Database Access Language is a language used to write commands to access, upsert, and delete data stored in a database.
- Users can write commands or query the database using Database Access Language before submitting them to the database for execution.
- Through utilizing the language, users can create new databases and tables, insert data and delete data.
- Examples of database languages are SQL (structured query language), My Access, Oracle, etc. A database language is comprised of two languages.

2. Elaborate the types of a database management system with neat diagram.

DBMS is a collection of programs used for managing data and simultaneously it supports different types of users to create, manage, retrieve, update and store information.

Types of DBMS

The types of DBMS based on data model are as follows −

- Relational database.
- Object oriented database.
- Hierarchical database.
- Network database.

Relation Database

A relational database management system (RDBMS) is a system where data is organized in two-dimensional tables using rows and columns.

This is one of the most popular data models which is used in industries. It is based on SQL.

Every table in a database has a key field which uniquely identifies each record.

This type of system is the most widely used DBMS.

Relational database management system software is available for personal computers, workstation and large mainframe systems.

For example − Oracle Database, MySQL, Microsoft SQL Server etc.

Std ID	**Name**	**City**
201	Bob	Hyderabad
204	Lucky	Chennai
205	Pinky	Bangalore

In the above student table Std ID, Name and city are called as attributes and their values. Std ID is a primary key attribute which uniquely identifies each record in the student table.

Object Oriented Database

It is a system where information or data is represented in the form of objects which is used in object-oriented programming.

It is a combination of relational database concepts and object-oriented principles.

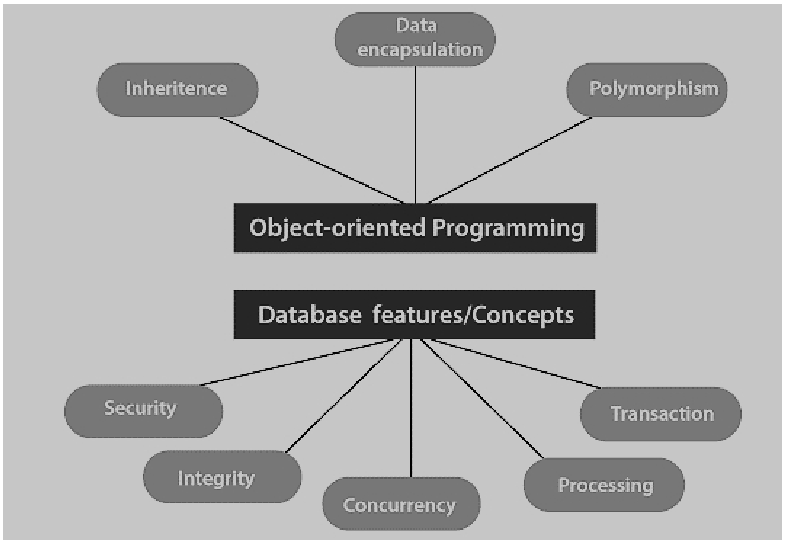
Relational database concepts are concurrency control, transactions, etc.

OOPs principles are data encapsulation, inheritance, and polymorphism.

It requires less code and is easy to maintain.

For example − Object DB software.

The object-oriented database is represented in diagram format below −



Hierarchical Database

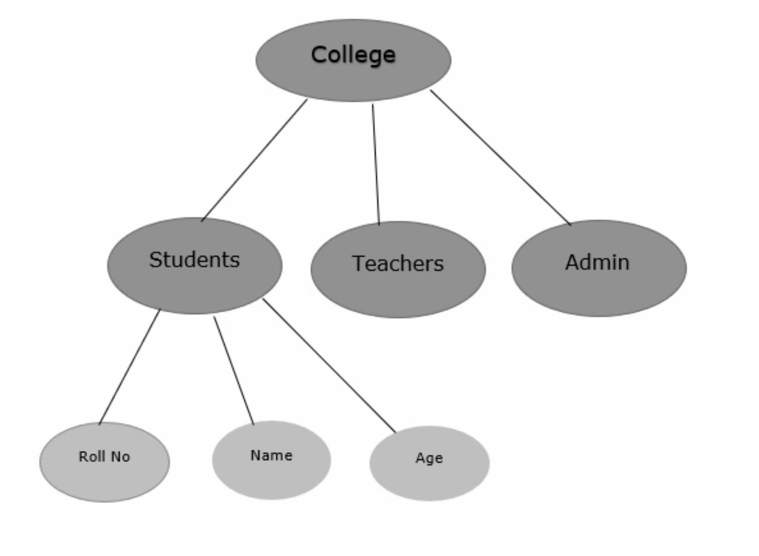
It is a system where the data elements have a one to many relationship (1: N). Here data is organized like a tree which is similar to a folder structure in your computer system.

The hierarchy starts from the root node, connecting all the child nodes to the parent node.

It is used in industry on mainframe platforms.

For example− IMS(IBM), Windows registry (Microsoft).

An example of a hierarchical database is given below −



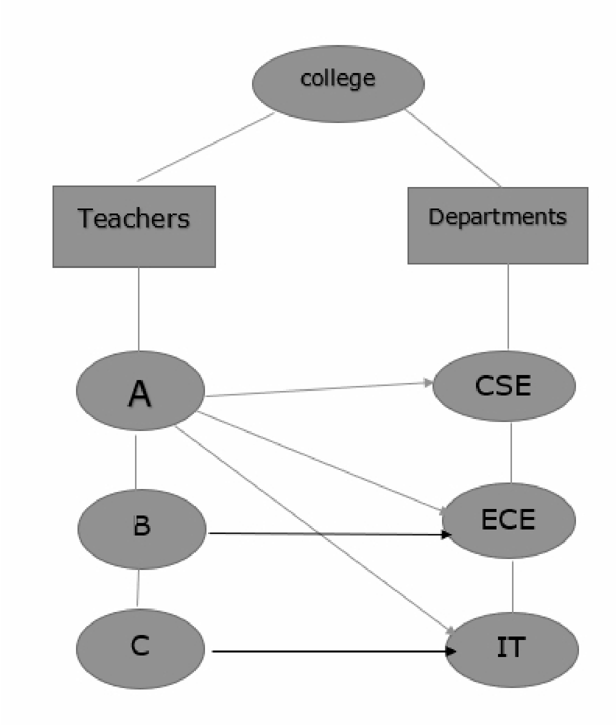
Network database

A Network database management system is a system where the data elements maintain one to one relationship (1: 1) or many to many relationship (N: N).

It also has a hierarchical structure, but the data is organized like a graph and it is allowed to have more than one parent for one child record.

Example

Teachers can teach in multiple departments. This is shown below −

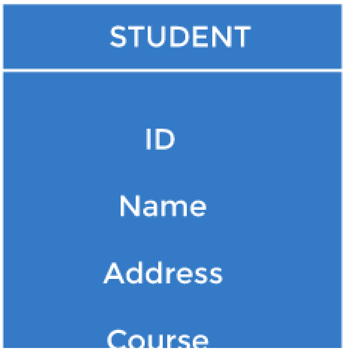


3. Explain the Keys used in Database Management System

Keys

- Keys play an important role in the relational database.
- It is used to uniquely identify any record or row of data from the table. It is also used to establish and identify relationships between tables.

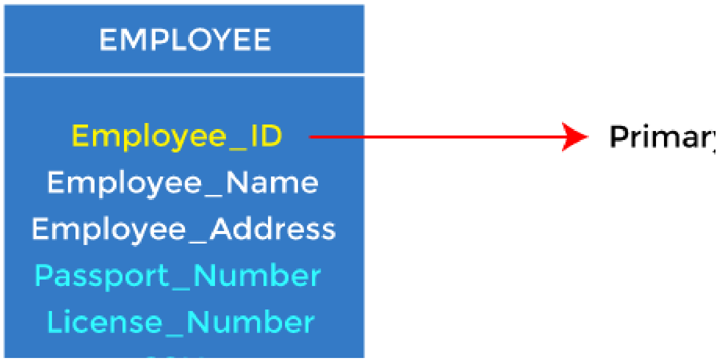
For example, ID is used as a key in the student table because it is unique for each student. In the PERSON table, passport_number, license_number, SSN are keys since they are unique for each person.



Types of keys:

1. Primary key

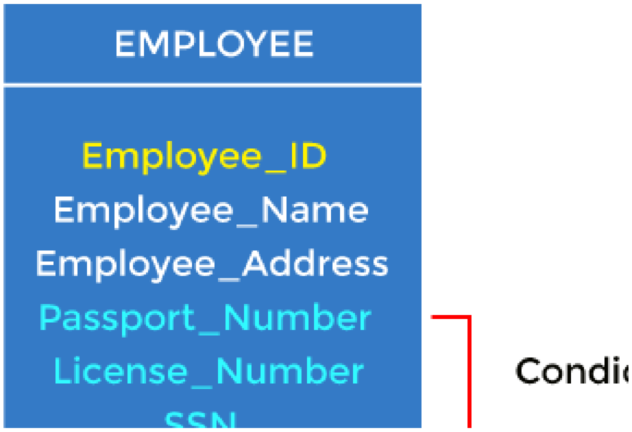
- It is the first key used to identify one and only one instance of an entity uniquely. An entity can contain multiple keys, as we saw in the PERSON table. The key which is most suitable from those lists becomes a primary key.
- In the EMPLOYEE table, ID can be the primary key since it is unique for each employee. In the EMPLOYEE table, we can even select License_Number and Passport_Number as primary keys since they are also unique.
- For each entity, the primary key selection is based on requirements and developers.



2. Candidate key

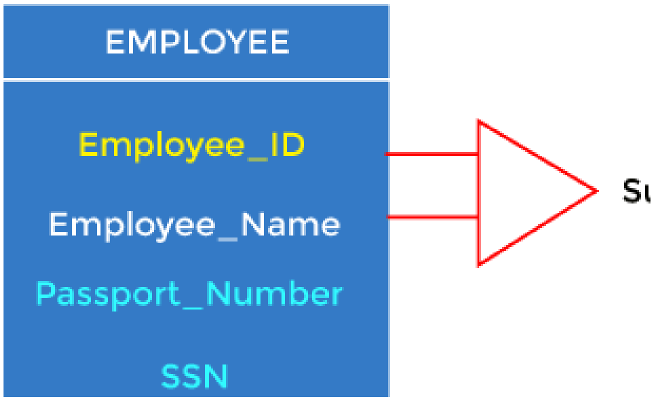
- A candidate key is an attribute or set of attributes that can uniquely identify a tuple.
- Except for the primary key, the remaining attributes are considered a candidate key. The candidate keys are as strong as the primary key.

For example: In the EMPLOYEE table, id is best suited for the primary key. The rest of the attributes, like SSN, Passport_Number, License_Number, etc., are considered a candidate key.



3. Super Key

Super key is an attribute set that can uniquely identify a tuple. A super key is a superset of a candidate key.



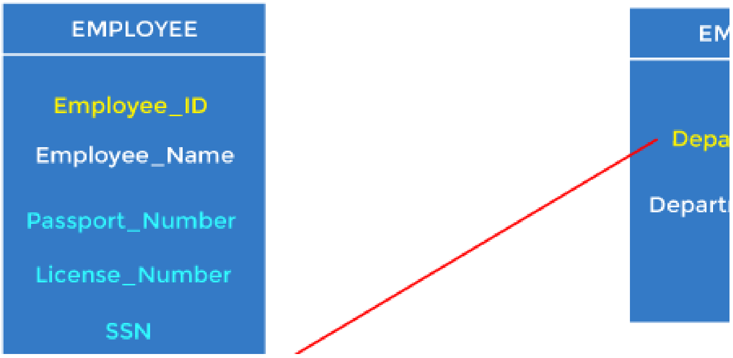
For example: In the above EMPLOYEE table, for(EMPLOEE_ID, EMPLOYEE_NAME), the name of two employees can be the same, but their EMPLYEE_ID can't be the same. Hence, this combination can also be a key.

The super key would be EMPLOYEE-ID (EMPLOYEE_ID, EMPLOYEE-NAME), etc.

4. Foreign key

- Foreign keys are the column of the table used to point to the primary key of another table.

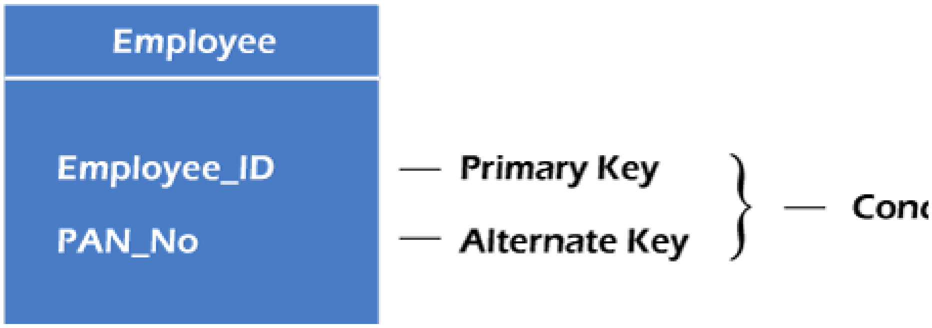
- Every employee works in a specific department in a company, and employee and department are two different entities. So we can't store the department's information in the employee table. That's why we link these two tables through the primary key of one table.
- We add the primary key of the DEPARTMENT table, Department_Id, as a new attribute in the EMPLOYEE table.
- In the EMPLOYEE table, Department_Id is the foreign key, and both the tables are related.



5. Alternate key

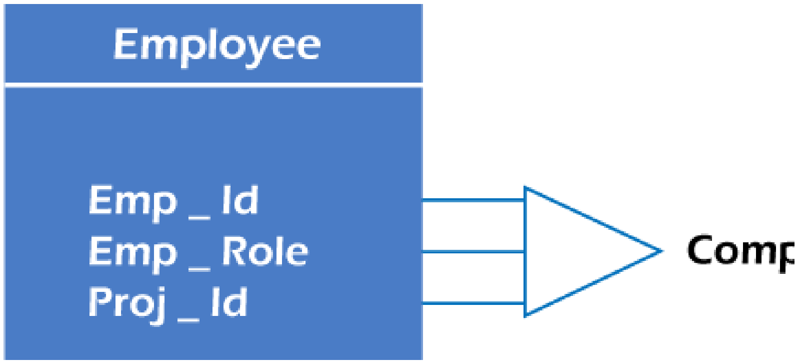
There may be one or more attributes or a combination of attributes that uniquely identify each tuple in a relation. These attributes or combinations of the attributes are called the candidate keys. One key is chosen as the primary key from these candidate keys, and the remaining candidate key, if it exists, is termed the alternate key. **In other words,** the total number of the alternate keys is the total number of candidate keys minus the primary key. The alternate key may or may not exist. If there is only one candidate key in a relation, it does not have an alternate key.

For example, employee relation has two attributes, Employee_Id and PAN_No, that act as candidate keys. In this relation, Employee_Id is chosen as the primary key, so the other candidate key, PAN_No, acts as the Alternate key.



6. Composite key

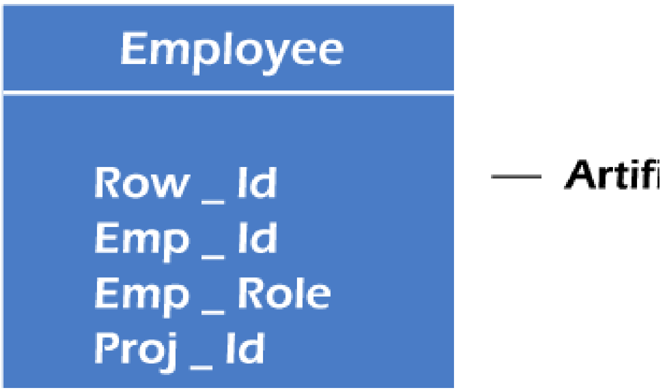
Whenever a primary key consists of more than one attribute, it is known as a composite key. This key is also known as Concatenated Key.



For example, in employee relations, we assume that an employee may be assigned multiple roles, and an employee may work on multiple projects simultaneously. So the primary key will be composed of all three attributes, namely Emp_ID, Emp_role, and Proj_ID in combination. So these attributes act as a composite key since the primary key comprises more than one attribute.

7. Artificial key

The key created using arbitrarily assigned data are known as artificial keys. These keys are created when a primary key is large and complex and has no relationship with many other relations. The data values of the artificial keys are usually numbered in a serial order.

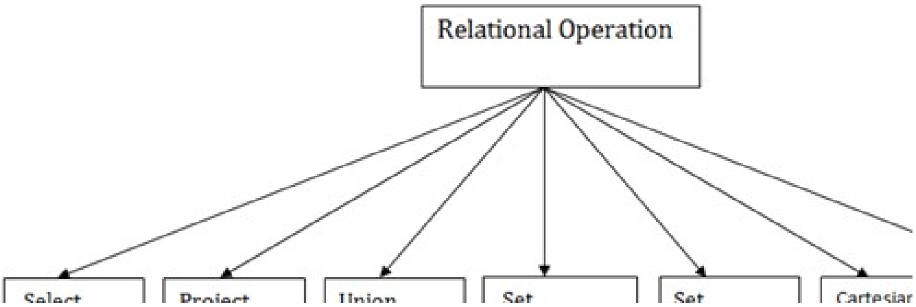


For example, the primary key, which is composed of Emp_ID, Emp_role, and Proj_ID, is large in employee relations. So it would be better to add a new virtual attribute to identify each tuple in the relation uniquely.

4. Elaborate on the Relational Algebra and its notation that can be applied banking sector.

Relational algebra is a procedural query language. It gives a step-by-step process to obtain the result of the query. It uses operators to perform queries.

Types of Relational operation



1. Select Operation:

- ○ The select operation selects tuples that satisfy a given predicate.
- ○ It is denoted by sigma (σ).

1. Notation: σ p(r)

Where:

σ is used for selection prediction

r is used for relation

p is used as a propositional logic formula which may use connectors like: AND OR and NOT.

These relational can use as relational operators like =, ≠, ≥, <, >, ≤.

For example: LOAN Relation

BRANCH_NAME	LOAN_NO	AMOUNT
Downtown	L-17	1000
Redwood	L-23	2000
Perryride	L-15	1500
Downtown	L-14	1500
Mianus	L-13	500
Roundhill	L-11	900
Perryride	L-16	1300

Query: σ BRANCH_NAME="perryride" (LOAN)

Output:

BRANCH_NAME	**LOAN_NO**	**AMOUNT**
Perryride	L-15	1500
Perryride	L-16	1300

2. Project Operation:

- This operation shows the list of those attributes that we wish to appear in the result. Rest of the attributes are eliminated from the table.
- It is denoted by ∏.

Notation: ∏ A1, A2, An (r)

Where, **A1**, **A2**, **A3** is used as an attribute name of relation **r**.

Example: CUSTOMER RELATION

NAME	**STREET**	**CITY**
Jones	Main	Harrison
Smith	North	Rye
Hays	Main	Harrison
Curry	North	Rye
Johnson	Alma	Brooklyn
Brooks	Senator	Brooklyn

Query: ∏ NAME, CITY (CUSTOMER)

Output:

NAME	**CITY**
Jones	Harrison
Smith	Rye
Hays	Harrison
Curry	Rye
Johnson	Brooklyn
Brooks	Brooklyn

3. Union Operation:

- Suppose there are two tuples R and S. The union operation contains all the tuples that are either in R or S or both in R & S.
- It eliminates the duplicate tuples. It is denoted by ∪.

Notation: R ∪ S

A union operation must hold the following condition:

- R and S must have the attribute of the same number.
- Duplicate tuples are eliminated automatically.

Example:

DEPOSITOR RELATION

CUSTOMER_NAME	ACCOUNT_NO
Johnson	A-101
Smith	A-121
Mayes	A-321
Turner	A-176
Johnson	A-273
Jones	A-472
Lindsay	A-284

BORROW RELATION

CUSTOMER_NAME	LOAN_NO
Jones	L-17
Smith	L-23
Hayes	L-15
Jackson	L-14
Curry	L-93
Smith	L-11
Williams	L-17

Query ∏ CUSTOMER_NAME (BORROW) ∪ ∏ CUSTOMER_NAME (DEPOSITOR)

Output:

CUSTOMER_NAME
Johnson
Smith
Hayes
Turner
Jones
Lindsay
Jackson
Curry
Williams
Mayes

4. Set Intersection:

- Suppose there are two tuples R and S. The set intersection operation contains all tuples that are in both R & S.
- It is denoted by intersection ∩.

Notation: R ∩ S

Example: Using the above DEPOSITOR table and BORROW table

Query: ∏ CUSTOMER_NAME (BORROW) ∩ ∏ CUSTOMER_NAME (DEPOSITOR)

Output:

CUSTOMER_NAME
Smith
Jones

5. Set Difference:

- Suppose there are two tuples R and S. The set intersection operation contains all tuples that are in R but not in S.
- It is denoted by intersection minus (-).

Notation: R - S

Example: Using the above DEPOSITOR table and BORROW table

Query: ∏ CUSTOMER_NAME (BORROW) - ∏ CUSTOMER_NAME (DEPOSITOR)

Output:

CUSTOMER_NAME
Jackson
Hayes
Willians
Curry

6. Cartesian product

- The Cartesian product is used to combine each row in one table with each row in the other table. It is also known as a cross product.
- It is denoted by X.

Notation: E X D

Example:

EMPLOYEE

EMP_ID	EMP_NAME	EMP_DEPT
1	Smith	A
2	Harry	C
3	John	B

DEPARTMENT

DEPT_NO	DEPT_NAME
A	Marketing
B	Sales
C	Legal

Query: EMPLOYEE X DEPARTMENT

Output:

EMP_ID	EMP_NAME	EMP_DEPT	DEPT_NO	DEPT_NAME
1	Smith	A	A	Marketing
1	Smith	A	B	Sales

1	Smith	A	C	Legal
2	Harry	C	A	Marketing
2	Harry	C	B	Sales
2	Harry	C	C	Legal
3	John	B	A	Marketing
3	John	B	B	Sales
3	John	B	C	Legal

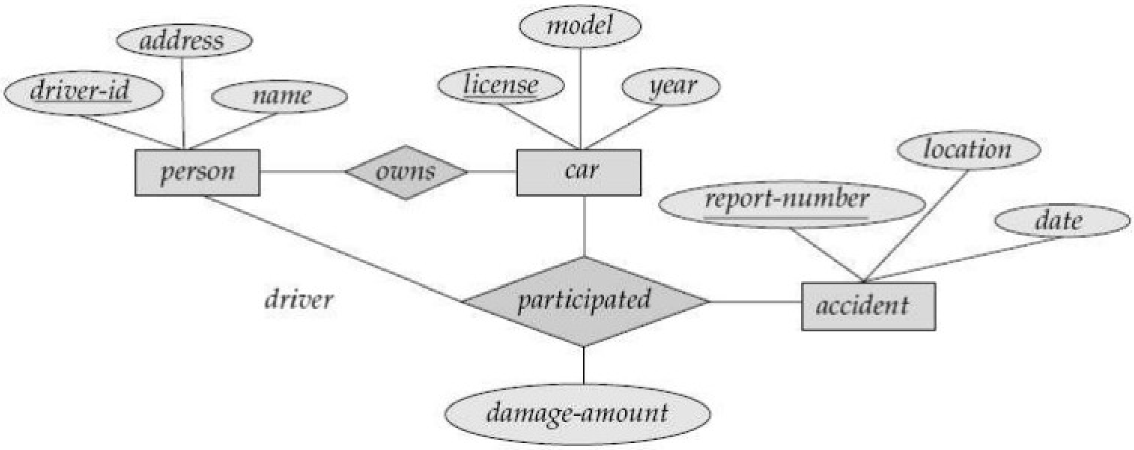
7. Rename Operation:

The rename operation is used to rename the output relation. It is denoted by **rho** (ρ).

Example: The rename operator to rename STUDENT relation to STUDENT1.

Query: ρ(STUDENT1, STUDENT)

5. Construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents

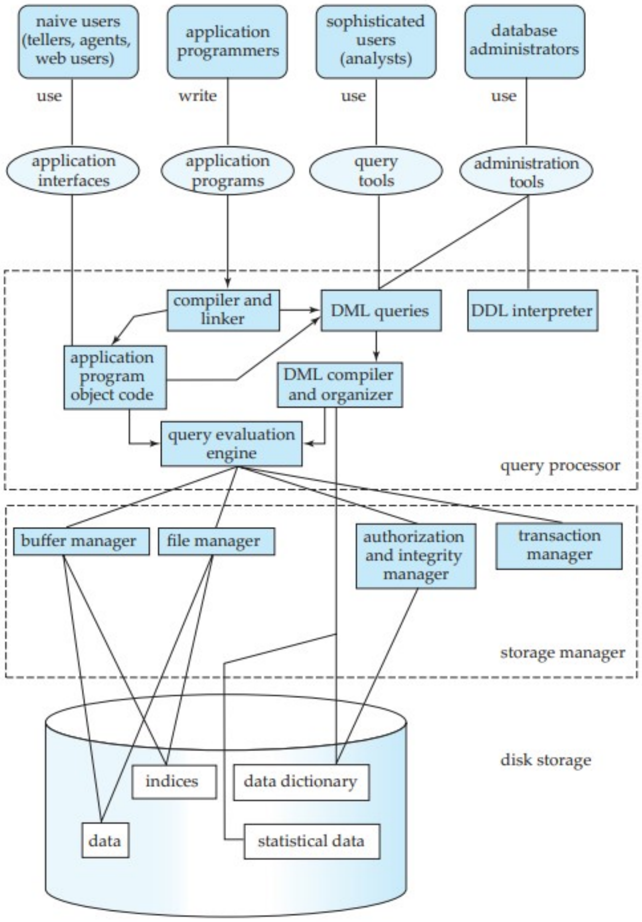


6. Explain the database system architecture with neat diagram?

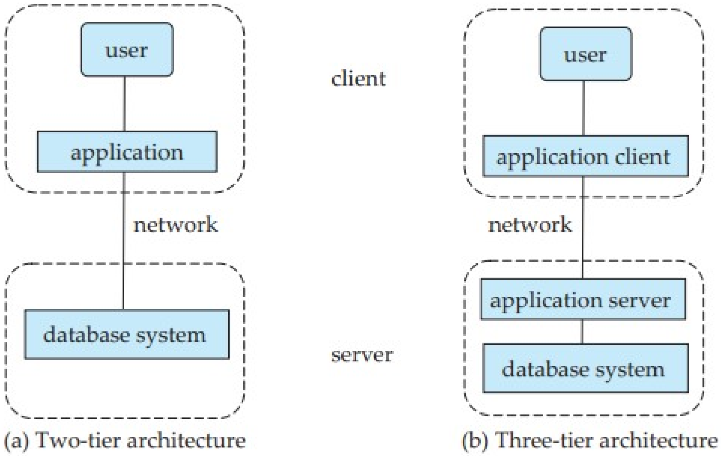
The architecture of a database system is greatly influenced by the underlying computer system on which the database system runs. Database systems can be centralized, or client-server, where one server machine executes work on behalf of multiple client machines. Database systems can also be designed to exploit parallel computer architectures. Distributed databases span multiple geographically separated machines.

Most users of a database system today are not present at the site of the database system, but connect to it through a network. We can therefore differentiate between client machines, on which remote database users work, and server machines, on which the database system runs.

System Structure



Two-tier and three-tier architectures



Database applications are usually partitioned into two or three parts. In a two-tier architecture, the application resides at the client machine, where it invokes database system functionality at the server machine through query language statements. Application program interface standards like ODBC and JDBC are used for interaction between the client and the server. In contrast, in a three-tier architecture, the client machine acts as merely a front end and does not contain any direct database calls. Instead, the client end communicates with an application server, usually through a forms interface. The application server in turn communicates with a database system to access data. The business logic of the application, which says what actions to carry out under what conditions, is embedded in the application server, instead of being distributed across multiple clients. Three-tier applications are more appropriate for large applications, and for applications that run on the World Wide Web.

7. Define relational algebra. Explain various relational algebraic operations with example

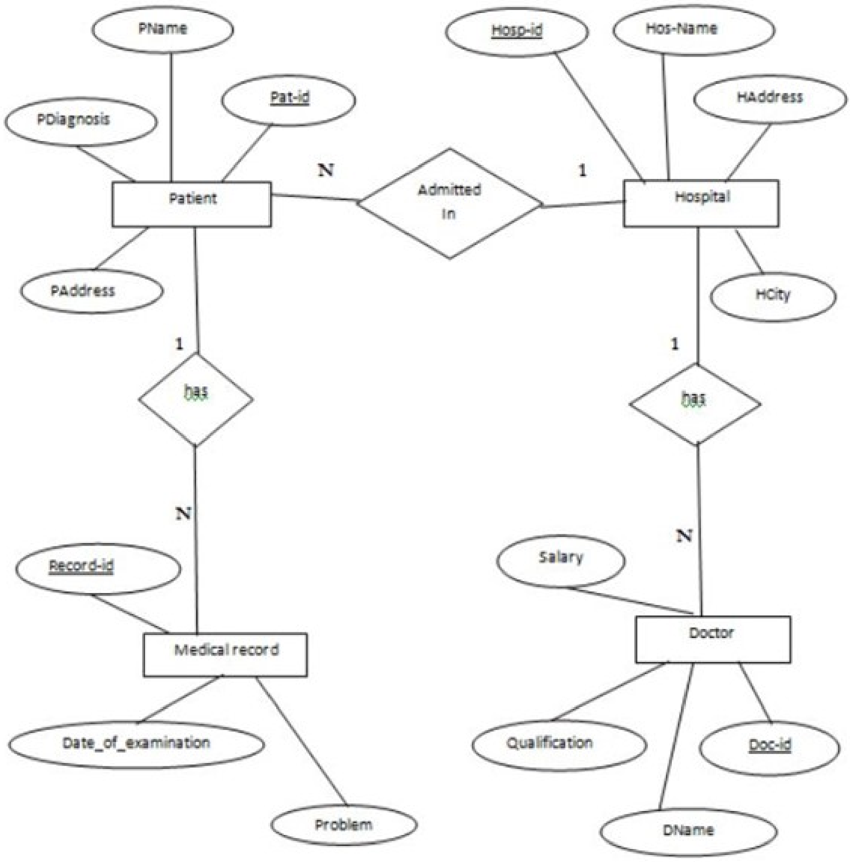
REFER Q.No 04 in PART B 13 Marks

8. Explain DDL with your own example?

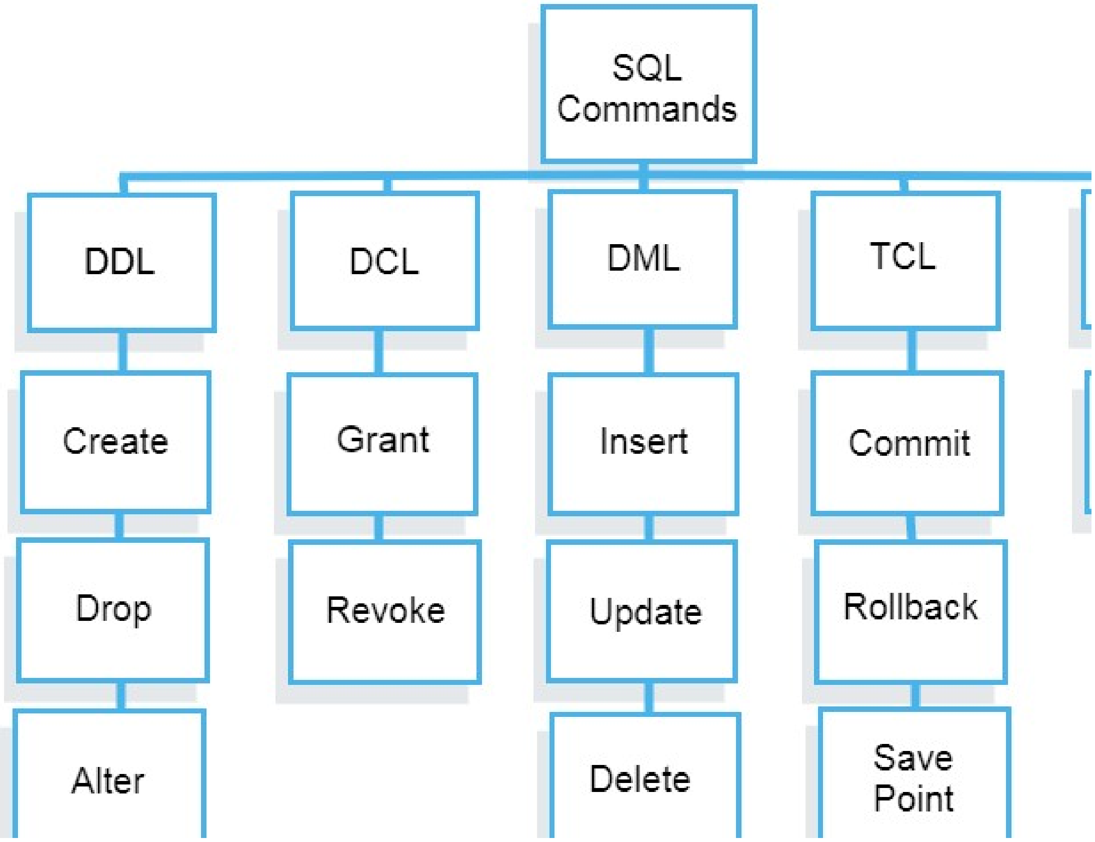
REFER Q.No 02 in PART B 14 Marks

PART B- 14 Marks

1. Construct the ER Diagram for Hospital Management system takes places the Four entity set (Patient, Doctor, Hospital, Medical record)



2. Examine on DDL, DML, TCL query for Blood Bank Information System.



DDL (Data Definition Language):

Data Definition Language is used to define the database structure or schema. DDL is also used to specify additional properties of the data. The storage structure and access methods used by the database system by a set of statements in a special type of DDL called a data storage and definition language. These statements define the implementation details of the database schema, which are usually hidden from the users. The data values stored in the database must satisfy certain consistency constraints.

Commands

CREATE: to create objects in database

ALTER: alters the structure of database

DROP: delete objects from database

RENAME: rename an objects

DML (Data Manipulation Language) :

DML statements are used for managing data with in schema objects.

DML are of two types –

Procedural DMLs: require a user to specify what data are needed and how to get those data.

Declarative DMLs (also referred as **Non-procedural DMLs**) : require a user to specify what data are needed without specifying how to get those data.

Declarative DMLs are usually easier to learn and use than procedural DMLs. However, since a user does not have to specify how to get the data, the database system has to figure out an efficient means of accessing data.

Commands:

SELECT: retrieve data from the database

INSERT: insert data into a table

UPDATE: update existing data within a table

DELETE: deletes all records from a table, space for the records remain

TCL (Transaction Control Language) :

Transaction Control Language commands are used to manage transactions in the database. These are used to manage the changes made by DML-statements. It also allows statements to be grouped together into logical transactions.

Commands

COMMIT: Commit command is used to permanently save any transaction into the database.

ROLLBACK: This command restores the database to last committed state.

It is also used with savepoint command to jump to a savepoint
in a transaction.

SAVEPOINT: Savepoint command is used to temporarily save a transaction so that you can rollback to that point whenever necessary.

Example:

```
CREATE TABLE Patient
   (Patient_ID         INT         NOT NULL,
    Patient_Name       VARCHAR(30) NOT NULL,
    Blood_Group        VARCHAR(4)  NOT NULL,
    Disease            CHAR(2)
            CHECK (Disease IN('A','B','AB','O')),
    Patient_Contact    NUMBER(10,0),
    Patient_Address    VARCHAR(45),
CONSTRAINT Patient_PK PRIMARY KEY (Patient_ID));

CREATE TABLE Donor
   (Donor_ID          INT        NOT NULL,
    Donor_Name        VARCHAR(30) NOT NULL,
    Medical_Report    VARCHAR(20) NOT NULL
            CHECK (Medical_Report IN('Satisfactory','Not Satisfactory')),
    Donor_Addresss    VARCHAR(45),
    Donor_Contact     NUMBER(10,0),
CONSTRAINT Donor_PK PRIMARY KEY (Donor_ID));

CREATE TABLE Blood_Bank
   (Blood_Bank_ID     INT           NOT NULL,
    Blood_Bank_Name    VARCHAR(50)    NOT NULL,
    Blood_Bank_Address VARCHAR(45),
    Blood_Bank_Contact Number(10,0),
    Donor_ID          INT,
    Patient_ID        INT,
CONSTRAINT Blood_Bank_PK PRIMARY KEY (Blood_Bank_ID),
CONSTRAINT Blood_Bank_FK1 FOREIGN KEY (Donor_ID)  REFERENCES Donor(Donor_ID),
CONSTRAINT Blood_Bank_FK2 FOREIGN KEY (Patient_ID) REFERENCES
Patient(Patient_ID));

INSERT INTO Patient VALUES (101,'Jose Adams','AB+','O',7778889999,'101 Melborne') ;
INSERT INTO Patient VALUES (102,'Rose Clara','B+','AB',8889999777,'102 Austria') ;
INSERT INTO Patient VALUES (103,'Ray Spencer','0-','A',6669999777,'103 Naple') ;
INSERT INTO Patient VALUES (104,'Sara Crux','AO','B',8885559777,'104 Paris') ;
INSERT INTO Patient VALUES (105,'Nate Zara','A+','O',1119999777,'105 London') ;

INSERT INTO Donor VALUES (201,'Lara Croft','Satisfactory','201 Meru',1112223333);
INSERT INTO Donor VALUES (202,'Manuel Ladder','Not Satisfactory','202 Lamu',2223334444);
INSERT INTO Donor VALUES (203,'Zain Orange','Satisfactory','203 Nairobi',3334445555);
INSERT INTO Donor VALUES (204,'Anne Gibbs','Not Satisfactory','204 Narok',4445556666);
INSERT INTO Donor VALUES (205,'Reign Stomy','Satisfactory','205 Vasha',5556667777);
INSERT INTO Blood_Bank Values (301,'Meru Blood Bank','301 Meru','6667778888',201,101);
INSERT INTO Blood_Bank Values (302,'Lamu Blood Bank','302 Lamu','7778889999',203,102);
INSERT INTO Blood_Bank Values (303,'Nairobi Blood Bank','303 Nairobi','8889991111',205,103);
INSERT INTO Blood_Bank Values (304,'Narok Blood Bank','304 Narok','9991112222',205,104);
INSERT INTO Blood_Bank Values (305,'Vasha Blood Bank','305 Vasha','1112223333',203,105);

SELECT Patient_Name,Blood_Group,Disease
FROM Patient
WHERE Patient_ID = 101;

SELECT Donor_Name,Donor_Addresss,Donor_Contact,Medical_Report
FROM Donor
WHERE Medical_Report = 'Satisfactory';

SELECT Patient_Name,Blood_Group,Blood_Bank_Name,blood_bank.patient_id,patient.patient_id
FROM Patient, Blood_Bank
WHERE blood_bank.patient_id = patient.patient_id;

SELECT Donor_Name,Medical_Report,Donor_Addresss,donor.donor_id,Blood_Bank_Name
FROM Donor,Blood_Bank
WHERE blood_bank.donor_id = donor.donor_id;

SELECT Blood_Bank_Name, Blood_Bank_Contact,Patient_Name,Patient.Patient_id
FROM Blood_Bank, Patient
WHERE patient.patient_id = blood_bank.patient_id;

SELECT Patient_Name, Donor_Name, Blood_Bank_Name, Blood_Group, Medical_Report
FROM Blood_Bank,Patient,Donor
WHERE blood_bank.patient_id = patient.patient_id AND blood_bank.donor_id = donor.donor_id;

SELECT Donor_Name, Donor.Donor_ID, Blood_Bank.Donor_ID
FROM Donor LEFT OUTER JOIN  Blood_Bank
ON Blood_Bank.Donor_ID = donor.donor_id
GROUP BY blood_bank.donor_id;
```

3. Consider the following relational schema Employee (empno, name, office, age) Books (isbn, title, authors, publisher) Loan (empno, isbn, date) Write the following queries in relational algebra.

a. Find the names of employees who have borrowed a book Published by McGraw-Hill?

b. Find the names of employees who have borrowed all books Published by McGraw-Hill?

c. Find the names of employees who have borrowed more than five different books published by McGraw-Hill?

d. For each publisher, find the names of employees who have borrowed?

a. π_{name} (employee $\bowtie$ (π_{empno} (Loan $\bowtie$ (π_{isbn}($\sigma_{publisher='McGrawHILL'}$ (books)))))

b. π name, isbn(employee $\bowtie$ π_{empno} (loan $\bowtie$books) $\div$ π isbn($\sigma_{publisher='McGrawHILL'}$ (books))

c. π name (employee $\bowtie$ π_{empno} loan $\bowtie$books (π isbn($\sigma_{publisher='McGrawHILL' \hat{}count(isbn)>5}$(books))

d. π name (employee $\bowtie$ π_{empno} loan $\bowtie$books (π isbn($\sigma_{publisher='McGrawHILL' \hat{}count(isbn)>5}$(books))

4. Let E1 and E2 be two entities in an E/R diagram with simple single-valued attributes. R1 and R2 are two relationships between E1 and E2, where R1 is one to-many and R2 is many-to-many. R1 and R2 do not have any attributes of their own. Calculate the minimum number of tables required to represent this situation in the relational model?

Strong entities E1 and E2 are represented as separate tables. In addition to that many-to-many relationships(R2) must be converted as separate table by having primary keys of E1 and E2 as foreign keys. One-to-many relationship (R1) must be transferred to \'many\' side table(i.e. E2) by having primary key of one side(E1) as foreign key(this way we need not to make a separate table for R1). Let relation schema be E1(a1,a2) and E2(b1,b2). Relation E1(a1 is the key)

a1 a2

----

1 3

2 4

3 4

Relation E2(b1 is the key, a1 is the foreign key, hence R1(one-many) relationship set satisfy here)

b1 b2 a1

----

7 4 2

8 7 2

9	7	3

Relation R2 ({a1, b1} combined is the key here , representing many-many relationship R2)

a1	b1
1	7
1	8
2	9
3	9

Hence, we will have minimum of 3 tables.